

**Using Natural Language Processing to Analyse the Musical Lyric Decency of Malaysia's
Top Hit Songs**

by

Faridul Hakim Bin Farhan

16005195

Dissertation submitted in partial fulfilment of

the requirements for the

Bachelor of Information Technology (Hons)

September 2021

Universiti Teknologi PETRONAS

32610 Bandar Seri Iskandar

Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Using Natural Language Processing to Analyse the Musical Lyric Decency of Malaysia's
Top Hit Songs

by

Faridul Hakim Bin Farhan

16005195

A project dissertation submitted to the

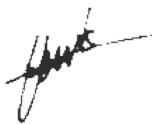
Information Technology Programme

Universiti Teknologi PETRONAS

in partial fulfilment of the requirement for the

BACHELOR OF INFORMATION TECHNOLOGY (Hons)

Approved by,



(Dr Yew Kwang Hooi)

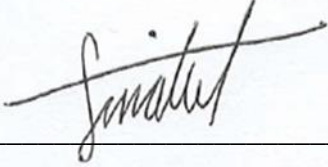
UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR, PERAK

September 2021

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



FARIDUL HAKIM BIN FARHAN

ABSTRACT

This project focuses on detecting sensitive content within song lyrics of Malaysia's top rated songs by implementing Natural Language Processing (NLP). This project requires the research and development in both "Explicit Content" exploration and analysis as well as "Sentiment Analysis" in both the Malay and English language. Malaysians love listening to music in multiple languages however this project intends to focus only on the English and Malay languages. As of currently, there are no publicly available datasets that can be used for the project which means that a dataset has that meets the requirements of the project needs to be created beforehand. The research aims to grab Malaysia's most popular songs currently being listened to, develop a machine learning model that can score the songs' "decency" value and produce an easy to understand grading system. This project utilized the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology as the main focus was building a machine learning model and CRISP-DM is the most efficient methodology to develop a machine learning model. CRISP-DM emphasizes on the business understanding of a project of knowing what people wish to see as an outcome of the project which would be used as a success indicator. This project was able to develop a machine learning model that is able to accurately predict and score songs according to the lyric content with an accuracy score of 81.21% which is higher to similar work done using Malay tweets. Although the results were satisfying, further enhancing the datasets used in multiple sections of the project would yield better results.

ACKNOWLEDGEMENTS

First and foremost, I would extend my gratitude towards Universiti Teknologi PETRONAS (UTP) for providing me with the best environment they could to help me grow into becoming a more knowledgeable, skilful and competent person. I am grateful for all the opportunities and resources UTP has offered to me since my first day of my foundation year here to the final day I spend here before I graduate. UTP has provided me with not only adequate resources, but reliable lecturers that I have taught me and guided me on how to not only deal with problems academically but also in general life. I especially thank UTP for preparing me with real work experience during the internship period they include in their educational structure which helped me a lot in preparing to join the work force.

I would also like to extend my gratitude towards my supervisor, Dr Yew Kwang Hooi for suggesting me the research topic that this paper is about. His suggestions lead me to explore topics that have interested me for a while in regards to data mining, processing and even machine learning. His suggestion also has lead me to explore more ideas on how I should approach my research topic.

Next, I would like to thank my friends who have assisted me in completing this project by giving me ideas and sometimes even solutions to the issues I faced while conducting the project. I am very much grateful for the help of my friends who taught me how to look for the solutions and provided me with packages and libraries that really helped me finish this research.

Lastly, I would like to thank the 43 respondents who replied to my survey as the findings helped me get a grasp on how relevant my project is as well as where should I focus my project towards.

Table of Contents

CERTIFICATION OF APPROVAL	ii
CERTIFICATION OF ORIGINALITY	iii
ABSTRACT.....	iv
ACKNOWLEDGEMENTS.....	v
CHAPTER 1: INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Objectives and Scope of Study.....	2
CHAPTER 2: LITERATURE REVIEW	4
2.1 Profanity and Its Affects to Their Users and Their Surroundings	4
2.2 NLP Development Challenges for The Malay Language.....	5
2.3 Methods That Have Been Used to Develop NLP for Profanity Detection	6
2.4 The Actual Amount of Sensitive Content Appearing in Media.....	7
2.5 Music Effects to Their Young Listeners	8
CHAPTER 3: METHODOLOGY	10
3.1 Methodology	10
3.1.1 Business Understanding.....	10
3.1.2 Data Understanding	10
3.1.3 Data Preparation.....	13
3.1.4 Modeling	16
3.1.5 Evaluation	16
3.2 Project Work	19
3.2.1 Compiler	19
3.2.2 Language.....	20
3.3.3 Packages.....	20
3.3 Flow Chart	21
3.4 Gantt Chart.....	22

CHAPTER 4: RESULTS AND DISCUSSION.....	24
4.1 Results.....	24
4.2 Discussion.....	25
CHAPTER 5: CONCLUSION AND RECOMMENDATION	29
5.1 Conclusion	29
5.2 Recommendation	29
References.....	31

List of Figures:

Figure 1: CRISP-DM Development Cycle	10
Figure 2: "billboard.com" Logo	11
Figure 3: "popnable.com" Logo	11
Figure 4: English Song Title and Artist Web Scraping Function	11
Figure 5: Malay Song Title and Artist Web Scraping Function	12
Figure 6: Song Lyrics Web Scraping Function.....	12
Figure 7: Song Labelling Function	13
Figure 8: Dropping "Nan" Values.....	13
Figure 9: Labeling Songs With Non-ASCII Characters	14
Figure 10: Rechecking The Songs That Had Non-ASCII Characters	14
Figure 11: Splitting the Into Train and Test.....	15
Figure 12: Tkenizing, Sequencing and Padding Data.....	15
Figure 13: Model Building.....	16
Figure 14: Model Building.....	16
Figure 15: 30 Epoch Model Training 1.....	17
Figure 16: 30 Epoch Model Training 2.....	17
Figure 17: 60 Epoch Model Training.....	17
Figure 18: Loading "Negaraku" for Model Accuracy	17
Figure 19: Predicting "Negaraku" Sensitive Content	18
Figure 20: Testing "Roy Jones- Can't be Touched" for Model Accuracy	18
Figure 21: Predicting "Roy Jones - Can't be Touched" Sensitive Content	18
Figure 22: Prediction Results.....	18
Figure 23: Scikit-learn Evaluation Calling	18
Figure 24: Accuracy Score, Confusion Matrix & Classification Report	19
Figure 25: SPYDER Logo	19
Figure 26: Python Logo	20
Figure 27: FLOW Chart of the Machine Learning Process.....	21
Figure 28: Pie Chart Representing Radio Channels People Listen to	25
Figure 29 : Pie Chart Representing Streaming Services People Listen to.....	25
Figure 30: Pie Chart Representing Language of Songs People Listen to	25
Figure 31: Pie Chart Representing the Frequency People Listen Music	26
Figure 32: Pie Chart Representing How Much Profanity People Use.....	26
Figure 33: Pie Chart Representing Which Version of Songs People Prefer to Listen to.....	26

Figure 34: Pie Chart Representing People’s Opinion of Music’s Impact on Their Behaviour	26
Figure 35: Pie Chart Representing People’s Opinion on Banning Certain Songs	27
Figure 36: Pie Chart Representing People’s Opinion of Lyrics Whether They Affect Their Fondness of The Song.....	27
Figure 37: Bar Graph of Total Score 20 Songs Rated By Respondents	27
Figure 38: Bar Graph of Mean, Median and Mod Scores of Songs Rated By Respondents ...	28

List of Tables:

Table 1: Study Outcome Comparison.....	7
Table 2: Confusion Matrix Explanation	19
Table 3: Table of Packages Used.....	20
Table 4: FYP1 Gantt Chart	22
Table 5: FYP2 Gantt Chart	23

CHAPTER 1: INTRODUCTION

1.1 Background

Listening to music is a common activity that many partake in their daily life nowadays due to modern technology making it available to almost everyone all the time everywhere. People listen to music on a variety of platforms including streaming services such as “Spotify”, “Apple Music” and “Youtube”, radio channels as well as more physical means such as the almost abandoned album compact disks (CD). Nowadays, these platforms can be accessed using a multitude of devices that are easily obtainable and used by most people such as smartphones, computers, smart tablets, car radios, mp3 players as well as disk readers. As of 2019 the International Federation of the Phonographic Industry (IFPI) has claimed that the average listener spends around 18 hours a week listening to music. Taking the statistic mentioned, the average person nowadays spends 10% of their time or life listening to music and although some might think that that number is small and insignificant, it is a number that has been increasing and is expected to increase further in the future. As of now, listening to music is a frequent activity participated by almost everyone, and many are wise to be worried if it may affect us the listeners’ behaviour and personality. In Malaysia specifically, we listen to a wide range of songs that cover various genres and more importantly languages. Most Malaysians tune in to songs that are in the language of their respective ethnicity mainly Malay, Chinese and Indian. Including the languages mentioned previously, Malaysians also enjoy listening to songs that are in English, Arabic and more recently Japanese and Korean. Modern artist often create music videos to accompany the songs they release which often further demonstrates the messages within the songs through strong visuals that viewers can enjoy watching. These music videos are usually released to Youtube which allows almost anyone to view them. Recently there is a trend of releasing songs that are categorized as “explicit” which often contains what many would agree to be “sensitive” or “should be censored” content which could include anything from profanity, sexual or violent content. The music videos accompanying these songs are often very graphically representative of the song violent them songs have violent showing videos and sexual songs have sexual content videos. More often than not, a “clean” version of a song is released on the radio and people often search and listen to the “explicit” version of the song and just like any youth from any country, Malaysian youth are being exposed to these material which has caused a concern whether they will be or have been affected by them or not.

1.2 Problem Statement

The Malaysian youth are very much exposed to songs that more often than not will affect their personality, behaviour and morality due to the fact that music covers a significant amount of their daily life. As of currently there is no clear age classification or recommendation that controls the release and usage of songs similar to how films are rated namely “Rated 18 and Above” (R)/(18), “Parental Guidance” (PG) and “General Viewing” (G). Currently, any efforts to classify songs require manual processing done individually where people listen to songs and decide for themselves whether it is appropriate or not. This process is very time consuming and people do not usually inform the public of their findings on a song.

Secondly, there are no publicly available datasets of songs that can be used immediately in case any project similar to this one wanted to be conducted. There are websites that rank songs based on popularity, but non provide them as downloadable datasets and most if not all only list out the song titles and artist but not lyrics. There is also the problem that most websites update their music charts and do not save the previous chart is case people want to see them

Third, websites that have a database of songs and lyrics not only do not provide downloadable datasets, they often have complex website structure that makes it near impossible to copy and paste the song info and lyrics let alone scrape using programming methods

1.3 Objectives and Scope of Study

The study aims to help people be more aware of the content Malaysian youth are being exposed to in hopes that further action can be taken from the information found.

The objectives of this project are as follows:

- To identify Malaysia’s most current top voted songs from popular radio websites using web scraping method
- To develop a machine learning model that is able to predict the content severity level of each song
- To evaluate the songs based on the cleanliness score and visualizing the results

The scope of the study covers Malaysia's most recent top rated songs according to reliable sources including the most popular streaming services used by Malaysians such as Spotify and Apple Music as well as popular radio channels such as Hot.fm who releases a weekly "top 40" chart which is a list of songs rated by Malaysians which indicates which ones are most popular and being listened to now. The study aims to focus on songs only on the Malay, Indonesian and English languages as those are the common languages listened by a large portion of listeners

CHAPTER 2: LITERATURE REVIEW

2.1 Profanity and Its Affects to Their Users and Their Surroundings

Profanity is understood and agreed by many to be words that are discouraged from being used by people who are seen as representing the social order such as teachers, parents and government officials due to often negative meanings the words may carry (DeFrank & Kahlbaugh, 2019). Most cultures have a sets of words or phrases that they categorize as “swear words” and although those words are publicly considered unsuitable for use, many partake in their usage as form expression. Often these words are used to deliver expressions of emotions which are likely to be negative such as anger or hurled towards someone or something as an insult. Profanity can best be understood as characterized by four categories: sacred, sex, bodily functions, and slurs. It was found the most people find that the most offensive obscene words were in relation to sex such as the word “fuck” while excretory words such as “ass” or “shit” are seen as less offensive is still considered very offensive. Different cultures and different languages have different profane words but most seem to agree anything that refers sexual activity, dirty and disgusting objects are categorized as profane. Another common ground that most cultures and languages agree upon is if a word is used as an insult, it is a profane word. An example of this is in English, “shit” is seen as a profane word. While it simply refers to excrement, many use the word to describe something as dirty or as an insult. While in Malaysia, the word “babi” refers to the pig, an animal but Malaysians see the word as dirty as it refers to a dirty animal and often used as an insult that degrades a person usually comparing them to indicating them that they are seen as dirty or stupid. Some religions heavily criticize the use of profanity and often relate it to an act of sinning

The usage of profanity often leads to people assuming those who use it as less intellectual and incompetent. More often than not, people who partake in swearing are seen as less capable and unappealing to interact with which leads to many avoid using swear words simply to make a good impression on others and the public. A study was conducted by Melanie DeFrank and Patricia Kahlbaugh in 2018, where 138 college students from a Northeast university in the United States ages ranged from 18 to 53 years with 101 women and 37 men was conducted to observe whether profanity affects impressions and judgements about the speaker. The research evaluated the participants’ reactions to the words used by each other when paired up in a few different settings. The conversation includes no profanity, one speaker using profanity, or both speakers using profanity, and the dyad includes either a same- or a mixed-gender pair. The study concluded that men and women rate the offensiveness of words very differently where

most women see the word “bitch” as being the most offensive word to say as it often used as derogatory term referring towards women (DeFrank & Kahlbaugh, 2019). The results indicated that the use of profanity leads to poorer impression rating regardless of the speaker’s gender or age. It also showed that the profane using speakers are rated lower on intelligence, trustworthiness, proneness to anger, deviancy, politeness, offensiveness, aggressiveness and likability. Many believe that people who use profanity have a small vocabulary and thus are unable to express themselves better although the belief not proven to be true. 45 of the 93 participants who received the profanity vignette, did not consider the language profane yet they rate others who use profanity with lower impression rate despite not considering the language as offensive. This leads to believe that perhaps they are unaware of their bias towards the topic and maybe is the reason why despite many people using it, the use of profanity is still looked down on by the public. While the gender of the speaker did not affect impressions, profanity usage in either same-gender or mixed-gender pair-ups did affect the sociability of the pairings. The study showed people are more comfortable using swear words in a same-gender environment rather than a mixed-gender environment.

2.2 NLP Development Challenges for The Malay Language

Developing any new NLP for a language can quite the difficult task due to the challenges that developers may face when attempting to design and create a model that is able to recognize the whole dictionary of a language. An example of a challenge is training a machine to understand a word that carries different meanings according to different context. A word can carry different meaning according to how a sentence is structured, where the word is placed or even what the word would be referring to. First of all, there are limited publicly available Malay text corpora as most are private or for academic usage (Lan & Logeswaran, 2020). It is believed that Dewan Bahasa and Pustaka, Pangkalan Data Korpus (UKM-DBP) is the current most complete and comprehensive available in regards to the Malay language currently but most researchers have resorted to compiling their own corpus using different methods such as using compiled tweets using Tweeter API.

Another challenge of developing NLP for the Malay language is from the fact that the Malay language itself is a borrow or even uses words that come from other language that could be anything from Arabic, Indian, Chinese or even English. The borrowed words often are written differently like the English word “police” is written as “polis” in Malay but there are some exceptions that could pose challenges. The biggest challenge would arise from new words that have not been integrated into the Malay language officially yet nor received a translation for

but is being used commonly by the public such as the term “selfie” which has a Malay term “swafoto” but is not commonly used. This could be a challenge specifically for this product as songwriters often write lyrics using words that are commonly used by the public which might not be included in any dictionary. Songwriters also often write their songs with abbreviated words that were created by them to rhyme with other lyrics but is a dissection of a proper word making it difficult to be recognized by a machine.

2.3 Methods That Have Been Used to Develop NLP for Profanity Detection

There has been a multitude of efforts to detect certain messages using NLP from different researches that could be used as reference for this project. The first research to be named is Sentiment analysis of informal Malay tweets with deep learning where a group of researches use a deep learning model to analyze the content of tweets that are in the Malay language and perform sentiment analysis on them. The reason they decided to conduct the research using a deep learning model on twitter is due to the fact that twitter produces an average of 250 million tweets per day in 2020 and the number of Malay users in 2019 was over 2.5 million and the number has risen since giving them a large amount of data to use for their research. Sentiment analysis (SA) is also known as opinion mining where a machine extract opinions based on message. The end goal of SA is to develop a machine learning model that is able to extract the attributes of a message namely the subject of a message, the polarity of the message usually between positive or negative, and opinion holder which is the entity of the message or expression (Ying et al., 2020). The project concluded with a success in creating a model that was able to recognize the message as whole and determine whether it was positive or negative. This give an indication that a model that could analyze song lyrics could use the same model as a reference to extract the same information negative or positive.

Researchers from Graduate School of Knowledge Service Engineering, KAIST, Daejeon, Republic of Korea conducted Explicit Content Detection in Music Lyrics Using Machine Learning back in 2018 which they included the various methods they explored in the report they published. They believed that music heavily impacts the growth of children Korea and suggested to develop a machine learning model that could detect explicit content from song lyrics and classify the song as explicit which could then be used as reference to which songs should be restricted to the younger age groups. They mainly used two models to conduct their research (Chin et al., 2018). The first model automatically determines Fail/Pass in lyrics by learning from previously labeled data without a profanity dictionary while the other one does the same with a profanity dictionary that was created by the researches. Surprisingly, the

research concluded that the model that had no profanity dictionary had performed better than the one that had the profanity dictionary. This result may affect the direction of this project in which the same procedure taken by this group might be used as reference for exploring Malay songs.

Project Title	Description	Outcome
Explicit Content Detection in Music Lyrics Using Machine learning	A study carried out by Chin et al. (2018). They developed and tested 6 different models using a combination between either using Adaboost or Bagging algorithms, using all vocabulary, selective vocabulary, with a TF score or without a TF score	The model which utilized bagging and selective vocabulary outperformed all other models they tested including one which utilizes a man-made profanity dictionary. Model 6 achieve a precision value of 97%
Sentiment Analysis of Informal Malay Tweets with Deep Learning	A study carried out by Ying et al. (2020). They developed a Sentiment Analysis CNN model for the Malay and English Language and compared the results to other similar works	Their model achieve an accuracy of 77.59% which exceeds similar work done in using the Indonesian Language

Table 1: Study Outcome Comparison

2.4 The Actual Amount of Sensitive Content Appearing in Media

In order to find out relevancy of his project, how much profanity is prevalent in music must first be investigated to determine whether it is a concern or not in the first place. Previous research findings reveal that specific song genres in particular rap, hip hop and pop music which are particularly popular among young adolescents contain the most sensitive or even what some might consider as dangerous content (M. FRISBY & BEHM-MORAWITZ, 2019). The findings reveal that these songs that are categorized within these genres have a higher frequency of profanity, misogyny, violence and sexual messages contained in them. They also often tell stories or contain themes that are degrading or demeaning most of which are directed to women. Some share the opinion that these songs can be interpreted as audio pornography that endorses sexist and violent ideas and behaviours (M. FRISBY & BEHM-MORAWITZ, 2019).

A different research revealed that from pop song lyrics they on the Billboard chart of 2019 discovered that both male and female artist used profanities in their lyrics and that female artists use profanity to an equal or greater extent than male artists (you know). What differs the profanity usage between the two genders were the words more commonly used by each gender.

Young boys and girls are very much influenced by celebrities and the fact that some of these adored artists use profanity affects the speech and behaviour of their young fans. Most female artists inspire other women young and old alike to chase their dreams, be bold and outspoken in life and many young girls try copy famous female artists usually their fashion sense but most commonly their speech patterns. Just like how male artists who curse influence young boys to curse, the existence of female artists who curse influences young girls to curse.

2.5 Music Effects to Their Young Listeners

In order to investigate on the effect music has on its young listeners, a study that was published on a Health Science Journal back in 2018 was used as a reference. The study was focused on Dangdut Music and its effects on behaviour changes towards Indonesian adolescent youth. Dangdut music is an Indonesian unique song type or genre that are enjoyed by many Indonesians and some Malaysians even. The issue of Dangdut music that gives concern to many is the fact that most Dangdut song lyrics contain sexual and inappropriate words, phrases and messages that are often tied to impoliteness. Dangdut music is very popular among Indonesian youth as it often has easy to remember and straightforward lyrics that do not carry any hidden meaning that require analyzing and appreciation. Previous researches have proven that music does indeed affect the behaviour of its listeners as the intelligence and growth of children in relation to social and communication skills to emotions. Songs that contain sexual themes have been banned by the Indonesian Broadcasting Commission (KPI) due to the belief that the negative content might affect the behaviours and morality of listeners especially the adolescents but Dangdut music still prevails in the Indonesian music industry (Rahayu, 2018). Reports in 2013 revealed that some young male students danced while opening their trouser zippers in front of female students and some openly harass female students by taking off other female students' skirts. A study showed that most students who enjoy Dangdut music are going into their prepubescent or pubescent age which leads to them being easily influenced by their surroundings. This age group are very emotional, passionate, daring and impressionable. This age group are exploring their new interest in the opposite sex and their exposure to Dangdut music is believed to have exposed them to a skewed perception on acceptable or even appropriate interactions with the opposite sex causing them to often behave rudely toward them. The health journal went through seven different researches in relation to how exposure to Dangdut Music changes the behaviour of Indonesian adolescent youth and concluded children and teenagers often imitate what they hear or see especially if the subject is something that is trendy. Since Dangdut music is very popular especially among the adolescent youth, it

has become an invisible teacher that has affected their behaviour and morality. Most adolescent that listen to Dangdut music are seen as be very rude, speaking with vulgar words and some even harass women sexually.

CHAPTER 3: METHODOLOGY

3.1 Methodology

Cross-Industry Standard Process for Data Mining (CRISP-DM)

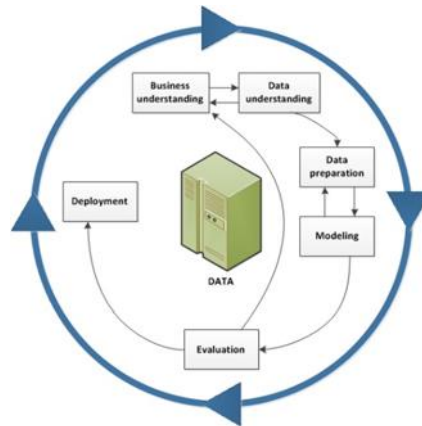


Figure 1: CRISP-DM Development Cycle

CRISP-DM is a process which describes the data science or data mining and processing life cycles. It is often used to guide data scientists on how they should organize and implement data into a machine learning project. The CRISP-DM methodology consist of six phases which are Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation and Deployment. For this project, the deployment phase was ignored as it was not a main concern of the project which is to build a model that is capable of meeting the objectives of the project.

3.1.1 Business Understanding

During this phase, I focused on understanding my project's objectives from a business perspective. The reason for doing so to achieve a clear understanding of what I want to see at the end of the project which is a list of popular songs in Malaysia and a simple to understand clean content grading system. This step is a foundational part of the project to have a clear indicator of what a "success" looks like. From here, I assessed the situation at hand and searched for the resources that I required which were a database of popular songs in Malaysia, a method of processing the lyrics of the songs and grading them, and a method of showing people the findings of my project.

3.1.2 Data Understanding

To ensure the success of the project, the most important resource needed is a database of songs with the songs' title, artist, lyrics and a label that indicates whether the song is "clean" and

“appropriate for children hearing” or not. After scouring the internet, I found no publicly available dataset that would meet the requirements of the project and opted to create my own dataset with actual values that could be found publicly.



Figure 2: "billboard.com" Logo



Figure 3: "popnable.com" Logo

Firstly, I had to obtain the song titles and artist of popular songs that Malaysians would listen to and since there are no available such list available, I decided to scrape information from two different websites “<https://www.billboard.com/charts/year-end/2020/hot-100-songs/>” (billboard.com) and “<https://popnable.com/malaysia/charts/top-40/year-2020>” (popnable.com) which provided the top 100 songs by the end of the years 2006 and 2020 as well as the top 200 songs in Malaysia by the end of the years 2016 and 2020 respectively. I decided to use these list due to the fact that Malaysians listen to songs from different languages but the most common ones listened to nearly all Malaysians were in English and Malay. The website billboard.com provided a list of songs enjoyed globally while popnable.com provides a list of songs produced and sung by Malaysian’s. I decided to scrape. I decided to only grab the song title and artist name for 100 songs between the years 2016 and 2020 from both websites to have an initial dataset of the same size for songs on both languages English and Malay.

```
def addlist(a):
    Songurl = (a)
    openSong = urlopen(Songurl)
    Song_soup = soup(openSong.read(), "html.parser")
    openSong.close()

    rawdatalist = Song_soup.findAll("li", {'class': 'o-chart-results'})

    i = 0
    raw_arr=[]
    firstSplit=[]
    while i<100:
        raw_arr.append(rawdatalist[i].text)
        firstSplit.append(raw_arr[i].split("\n\n\n"))
        placeholder1 = firstSplit[i][0].split("\n\n")
        placeholder2 = firstSplit[i][1].split("\n\n")
        song_list.append(placeholder1[1])
        artist_list.append(placeholder2[0])
        i+=1

    return song_list
    return artist_list
```

Figure 4: English Song Title and Artist Web Scraping Function

```

def addList(a):
    Songurl = a
    openSong = urlopen(Songurl)
    Song_soup = soup(openSong.read(), "html.parser")
    openSong.close()

    songs = Song_soup.findAll("a", {"class": "chart-song-name"})
    artists = Song_soup.findAll("a", {"class": "chart-singer"})
    i = 0
    songArr=[]
    artistArr=[]
    firstSongSplit=[]
    secondSongSplit=[]
    firstArtistSplit=[]
    secondArtistSplit=[]

    while i<20:
        songArr.append(songs[i].text)
        firstSongSplit.append(songArr[i].split("\r"))
        secondSongSplit.append(firstSongSplit[i][2].split("
song_list.append(secondSongSplit[i][1])
        artistArr.append(artists[i].text)
        firstArtistSplit.append(artistArr[i].split("\r"))
        secondArtistSplit.append(firstArtistSplit[i][2].split("
        artist_list.append(secondArtistSplit[i][1])
        i+=1

    return artist_list
    return song_list

```

Figure 5: Malay Song Title and Artist Web Scraping Function

I created a 2 python files to create the dataset to better organize my work. Each dataset had a defined function that was similar in which they both would receive a string input that was the link of the website to be scraped and would append two arrays that hold the song artists' names and the song titles. Both functions download the websites html file and searches the specific partition of the html that holds the song list and saves it as text in an array of string. Both websites have a different partition they use to construct and place their song list. The functions then split the strings and removes the unnecessary data such as "\n", "\r" and a long space in between. Afterwards, I combined the two datasets into a single DataFrame to be used for the lyrics data scraping. The reason I needed both the song artist and title was because some artist produced songs sharing a title and I required very specific reference to grab the lyrics for the song.

```

lyrics_list = []
def getLyrics(a):
    link = search(a)
    substring = "https://www.shazam.com/track/"
    if not link:
        lyrics_list.append("NaN")
        return lyrics_list
    else:
        if substring in link[0]:
            browser = webdriver.Firefox()
            browser.set_window_size(700,900)

            url = link[0]

            browser.get(url)
            time.sleep(4)

            html = browser.execute_script('return document.documentElement.outerHTML')
            html_soup = soup(html, "lxml")
            lyrics = html_soup.find("p", {"class": "Lyrics"})
            if lyrics is None:
                lyrics_list.append("NaN")
                browser.quit()
                return lyrics_list
            else:
                lyricsPass = lyrics.text
                browser.quit()
                lyrics_list.append(lyricsPass)
        else:
            lyrics_list.append("NaN")
            return lyrics_list

```

Figure 6: Song Lyrics Web Scraping Function

Next, I created another python file specifically to scrape song lyrics from “shazam.com” (shazam) a website that is recognized globally to store song lyrics from songs all around the world. In this file I created a function that searches a string that contains the artist name, song title and the string “https://www.shazam.com/track/” so that the first link from the search would be the correct shazam link of the song. The function will then check if the search produced any links, and if the first link is a shazam link. If the link is a shazam link, it then proceeds to open the link using a browser, download the whole html file and parses it into a readable html, searches for the html partition that holds the lyrics, appends the lyrics into an array of text and closes the browser. If any of the conditions are not met such as finding a link after searching, does not have a shazam link or does not have a lyric partition, the function appends with “NaN” value. The reason I decided to create a function that opens a browser to download the html file is because of a flaw in the package I used to web scrape the song titles and artists. The web scraping package I used, BeautifulSoup was incapable of downloading the whole html file from the shazam website and the solution was to use a different package, Selenium which could open web browsers and download the whole html file. The downloaded html file then needed to be parsed into readable html file using BeautifulSoup.

```
while i < len(DataSet):
    if any(x in DataSet.iloc[i][2].lower() for x in Profane_List):
        label.append(0)
    else:
        label.append(1)
    i+=1
```

Figure 7: Song Labelling Function

In order to label the songs, created a list of strings called “Profane_List” by combining two publicly available datasets of bad or cursed English phrases in their varied forms of writing from “https://github.com/surge-ai/profanity” and “https://www.kaggle.com/nicapotato/bad-bad-words” with a json file of my own which was filled with common bad phrases in the Malay language. I then created a function that goes through each lyric in the dataset and checks if it contains any phrases within the “Profane_List” list to label the songs with values “0” if true or “1” if false. This process however only takes after the Data Preparation phase.

3.1.3 Data Preparation

```
DataSet.dropna(axis=0,how="any",inplace=True)
DataSet.reset_index(inplace=True)
DataSet.drop(["index"],axis =1 ,inplace=True)
```

Figure 8: Dropping "Nan" Values

This phase prepares the dataset by viewing, modifying and selecting essential data for modeling. Before the data preparation process, the dataset consists of 896 rows. The first step in data preparation is removing rows with “NaN” within the dataset. As mentioned before in the Data Understanding phase, there is a possibility that the dataset contains some “NaN” values within the “lyrics” column due to either not meeting the requirements the function needed. The lyrics are essential to build an NLP model as it is used to train and test the model. There was a total of 107 “NaN” values in the dataset that needed to be removed. To remove the data, the dropna() function was called. The updated dataset then had its index reset so that it could be used properly

```
latin_Alphabet = []
def latin_Alphabet_Checker(s):
    return s.isascii()
i=0
while i < len(DataSet):
    lan = latin_Alphabet_Checker(DataSet.iloc[i][2])
    latin_Alphabet.append(lan)
    i+=1
DataSet['Is Latin'] = latin_Alphabet
```

Figure 9: Labeling Songs With Non-ASCII Characters

Afterwards, I noticed that some lyrics within the dataset was not in English or Malay. This is due to the fact that globally people have recently started to listen to Korean songs and some Malaysian artists include Indians and Chinese who write their songs in their respective languages. The previous lyric scraping function scraped the lyrics even though they are not American Standard Code for Information Interchange (ASCII) characters. I created a function that simply returns a Boolean value whenever a string is passed to it. I then passed the whole dataset through the function and appended the return value to an array. Any and all songs had any lyrics with non-ASCII characters returned the as FALSE

```
print(DataSet.loc[DataSet['Is Latin'] == False])
retry = DataSet.loc[DataSet['Is Latin'] == False]
retry.reset_index(inplace=True)
def recheck(a):
    return a.isascii() == False
i=0
while i < len(retry):
    ScndCheck = retry.iloc[i][3].split()
    if (sum(recheck(a) for a in ScndCheck) < (0.1*len(ScndCheck))):
        DataSet.at[retry.iloc[i][0], 'Is Latin'] = True
    i+=1
```

Figure 10: Rechecking The Songs That Had Non-ASCII Characters

After reviewing the data of songs that had non-ASCII characters, I noticed that many of the songs had a low non-ASCII character content such as American English songs using the word

“señorita” to refer to a woman but the whole song was filled with English words and regular ASCII characters. I decided it would be better if the songs that had a less than 10% content of non-ASCII characters were valid enough and usable for the dataset and created another function to recheck these specific songs. In this function, the failed songs go are checked one by one and the lyrics are spilt into singular word strings which are then counted. If the counted non-ASCII characters are less than 10% of the total characters of the song lyrics, the array is updated with a true value. Values that remained with false were dropped from the dataframe. At the end of the data culling, I was left with 777 rows left for the model training and testing.

```
import numpy as np
from sklearn.model_selection import train_test_split

y_dataset = Modeling_Data['Label']
x_dataset = Modeling_Data['Lyrics']

x = np.asarray(x_dataset)
y = np.asarray(y_dataset)

print(x[0])
print(y[0])

X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=27)
```

Figure 11: Splitting the Into Train and Test

After cleaning the data, I set the “x” and “y” values that are necessary for the machine learning. For the “x” value, I passed the whole “Lyrics” column while the “Label” column filled with “1” representing good and “0” representing bad was passed to the “y” value. The x and y value was then split randomly using scikit-learn method of splitting with a test size of 20% giving a ratio of 80% training and 20% testing data.

```
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from sklearn.model_selection import train_test_split

tokenizer = Tokenizer(oov_token="<OOV>")
tokenizer.fit_on_texts(X_train)
word_index = tokenizer.word_index
print(word_index)

training_sequences = tokenizer.texts_to_sequences(X_train)
print(training_sequences)

training_padded = pad_sequences(training_sequences, padding='post')
print(training_padded[0])

testing_sequences = tokenizer.texts_to_sequences(X_test)
testing_padded = pad_sequences(testing_sequences, padding='post')
```

Figure 12: Tkenizing, Sequencing and Padding Data

After splitting the data, the “X_train” data was then tokenized. The tokenizing of training data is necessary as to create a word index. A word index is a dictionary that stores and assigns every word string an index value which will then be used for the deep neural network(DNN)

model. Afterwards I turned the sentences into sequences of tokens and padded them. Sequencing tokens is simply structuring the tokens as if they were a sentence so that the machine learning model can see how sentences are structured. Padding is filling out the blank spaces of the token sequences so that all of them are the same length. For example, if the first sentence has 10 words and the second one has eight, padding will ensure both sentences have 10 words by filling the second sentence with “0”.

3.1.4 Modeling

```
vocab_size = 40000
embedding_dim = 16

model = tf.keras.Sequential([
    tf.keras.layers.Embedding(vocab_size, embedding_dim),
    tf.keras.layers.GlobalAveragePooling1D(),
    tf.keras.layers.Dense(24, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()
```

Figure 13: Model Building

The modelling phase is where I begin to build the machine learning model. First I set an embedding value of 16. Embedding is to give dimensions to the model of which to learn from. On a flat plane, there are only 2 dimensions that a result can fall into, x and y values. By giving the model bigger multi-dimensions, it is able to assign the training data to more plot areas. Words that appear only appear on the clean songs will appear on the far end of the clean direction and the opposite will fall in the opposite area and everything in between will slowly be filled by the model on its own. The more training data it has, the more it can try to plot the data and get better at properly assigning them. After it has been trained with adequate amount of data, it can summarize the vectors and give out a score of the cleanliness level of the song lyrics. “relu” and “sigmoid” are common DNN used for NLP.

3.1.5 Evaluation

```
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.summary()

history = model.fit(training_padded, y_train, epochs=30,
                    validation_data = (testing_padded, y_test))
```

Figure 14: Model Building

To evaluate the model, a few methods were used. The first method was by compiling and running the model showing the loss and accuracy on both training and testing data. The image above shows how I ran the model with an epoch of 30 and displayed the loss and accuracy of both the training and testing data

```
Epoch 1/30
20/20 [=====] - 1s 12ms/step - loss: 0.6849 - accuracy: 0.6232 - val_loss:
0.6772 - val_accuracy: 0.6346
Epoch 2/30
20/20 [=====] - 0s 8ms/step - loss: 0.6637 - accuracy: 0.6699 - val_loss:
0.6566 - val_accuracy: 0.6346
Epoch 3/30
20/20 [=====] - 0s 8ms/step - loss: 0.6365 - accuracy: 0.6731 - val_loss:
0.6336 - val_accuracy: 0.6410
```

Figure 15: 30 Epoch Model Training 1

```
Epoch 29/30
20/20 [=====] - 0s 8ms/step - loss: 0.1770 - accuracy: 0.9436 - val_loss:
0.4111 - val_accuracy: 0.8141
Epoch 30/30
20/20 [=====] - 0s 8ms/step - loss: 0.1662 - accuracy: 0.9581 - val_loss:
0.4129 - val_accuracy: 0.8141
```

Figure 16: 30 Epoch Model Training 2

The model summary show that at the beginning the training data had an accuracy score of 62% and the testing data had an accuracy score of 63%. After the model had run its full 30 epoch cycle it was able to learn and obtain a satisfying training accuracy of 95% and a testing accuracy of 81%. The testing accuracy is very important as it goes through words that have not been indexed and predicts using the training data.

```
Epoch 59/60
20/20 [=====] - 0s 8ms/step - loss: 0.0178 - accuracy: 1.0000 - val_loss:
0.5448 - val_accuracy: 0.7885
Epoch 60/60
20/20 [=====] - 0s 9ms/step - loss: 0.0167 - accuracy: 1.0000 - val_loss:
0.5474 - val_accuracy: 0.7821
[[0.9996262]]
[[0.00282231]]
Accuracy Score: 0.9472329472329473
```

Figure 17: 60 Epoch Model Training

As to why I decided to use an epoch value of 30 is because I had run the model with an epoch value of 60 and found that the model became overfitted where the training accuracy was 100% and the testing accuracy was at a 78%. I believe that the margin was too large and the model would be unusable. The 30 epoch model run however produce a realistic model with an accuracy margin of 6% only which i found to be the best value obtainable.

```
sentence_test_1 = [
    """"Negaraku, negaraku
    Kuberi sepenuhnya, kuberi sepenuhnya
    Ini negaraku oh darahku
    Hiduplah sepenuhnya dirgahayu semua
    Kalau kata itu kata kami hinga tamadun
```

Figure 18: Loading "Negaraku" for Model Accuracy

```

sequences1 = tokenizer.texts_to_sequences(sentence_test_1)
padded1 = pad_sequences(sequences1)
print(model.predict(padded1))

```

Figure 19: Predicting "Negaraku" Sensitive Content

```

sentence_test_2=[
    """Can't be touched
    Can't be stopped
    Can't be moved
    Can't be rocked
    Can't be shook
    We hot
    When will you niggas learn
    Came to get crunk
    Came to bring life

```

Figure 20: Testing "Roy Jones- Can't be Touched" for Model Accuracy

```

sequences2 = tokenizer.texts_to_sequences(sentence_test_2)
padded2 = pad_sequences(sequences2)
print(model.predict(padded2))

```

Figure 21: Predicting "Roy Jones - Can't be Touched" Sensitive Content

The second testing method I used is to register two songs of very different sensitive content level and run it through the already trained model to see if it is able to evaluate the songs. The first song is “Negaraku” Malaysia’s national patriotic song which should yield a high percentage, and the second song is Roy Jones’s “Can’t be Touched” which has a high count of profanity words in it that should result in a low test score.

```

Negaraku clean score : [[0.9418161]]
Roy Jones - Can't be Touched clean score : [[0.01353577]]

```

Figure 22: Prediction Results

The results are promising as the first song scored a high value of 94.18% which indicates that the song is almost fully clean, while the second song scored a 1.35% for having vary high bad content count.

```

y_pred = []
i=0
while i<len(Modeling_Data):
    y_pred_song = [Modeling_Data.iloc[i][2]]
    y_pred_seq = tokenizer.texts_to_sequences(y_pred_song)
    y_pred_pad = pad_sequences(y_pred_seq)
    y_tell = model.predict(y_pred_pad)
    if y_tell[0][0] > 0.5:
        y_pred.append(1)
    else:
        y_pred.append(0)
    i+=1

from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report

print('Accuracy Score: ', accuracy_score(y_true, y_pred))
print(confusion_matrix(y_true, y_pred))
print(classification_report(y_true, y_pred))

```

Figure 23: Scikit-learn Evaluation Calling

The last method used was using sklearn’s package to display the model’s overall accuracy, confusion matrix and classification report.

```

Accuracy Score: 0.8120978120978121
[[254  8]
 [138 377]]
precision  recall  f1-score  support
0         0.65   0.97   0.78     262
1         0.98   0.73   0.84     515

accuracy          0.81     777
macro avg         0.81     0.85     0.81     777
weighted avg      0.87     0.81     0.82     777
    
```

Figure 24: Accuracy Score, Confusion Matrix & Classification Report

	Actual 0	Predicted 1
Predicted 0	TN	FN
Actual 1	FP	TP

Table 2: Confusion Matrix Explanation

The results show that the model has an overall accuracy of 81.21% which is high in value and satisfactory for the model. The confusion matrix which represent values True Negative, False Negative, False Positive and True Positive in order of left to right and top to bottom indicated that the model was able to accurately predict 254 songs were truly negative in value or had a score closer to 0 while 377 songs were accurately predicted to be clean or having a value close to 1. For the classification report, the most important value to look at are the “recall” and “support”. The result classification report indicates that the model was able to accurately predict negative songs 97% of the time while positive songs only had a 73% prediction accuracy

3.2 Project Work

Tools Used

3.2.1 Compiler



Figure 25: SPYDER Logo

The Scientific Python Development Environment (SPYDER) is a free to use and open source developing platform and compiler that is able to use multiple languages and formats according to the developers' needs. SPYDER easily installs and loads new packages while also making it easier to view variables using their variable explorer and is able to plot and visualize data.

3.2.2 Language



Figure 26: Python Logo

Python is a general-purpose coding language that is widely used for machine learning and very suitable for Natural Language Processing.

3.3.3 Packages

1.	Pandas	A library used to create and manipulate dataframes for data exploration analysis and manipulation
2.	Numpy	A library used to support large multi-dimensional arrays and matrices
3.	Selenium	An open-source project that provides the tools necessary for web browser automation
4.	Scikit-learn	A machine learning library that offers a data splitting method and various classifications
5.	Tensorflow	A machine learning library commonly used to develop and train a deep neural network
6.	Keras	A library that provides Python interface for artificial neural networks and activation functions
7.	urllib	A package that collect modules required for URL related projects
8.	BeautifulSoup	A package for parsing html and xml documents in a readable and usable format to extract information from.
9.	googlesearch	A python library that searches "Google" and returns the links provided from the search.

Table 3: Table of Packages Used

3.3 Flow Chart

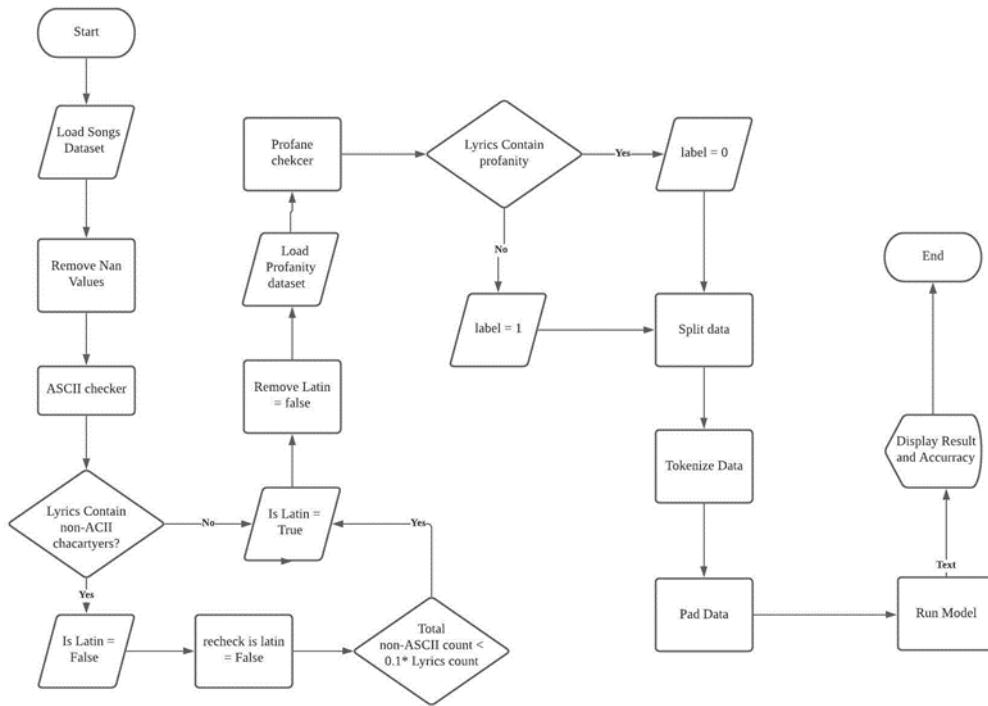


Figure 27: Flow Chart of the Machine Learning Process

3.4 Gantt Chart

FYP 1

Activities	1	2	3	4	5	6	7	8	9	10	11	12
Project Topic Selection												
Researching The Topic												
Data Analysis												
Progress Assessment 1												
Proposal Defense												
Interim Draft Report Submission												
Progress Assessment 2												
Interim Report Submission												

Table 4: FYPI Gantt Chart

FYP 2

Activities	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Gathering Data	█	█	█	█	█	█	█	█	█						
Cleaning Data	█	█	█	█	█	█	█	█	█						
Model Building							█	█	█	█	█				
Progress Assessment 1							█								
Dessertation Submission												█			
Demo Video Creation													█	█	
Demo Video Submission														█	
Progress Assessment 2													█		
VIVA															█

Table 5: FYP2 Gantt Chart

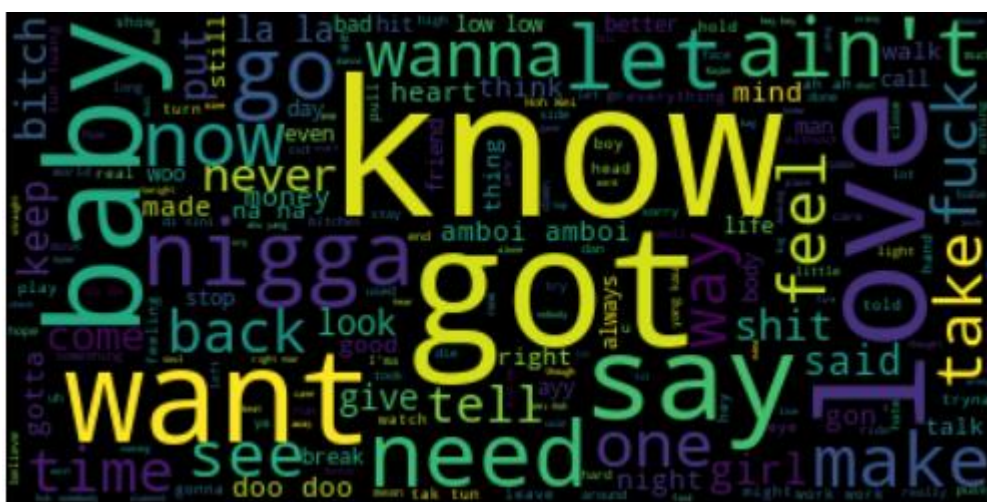
CHAPTER 4: RESULTS AND DISCUSSION

4.1 Results



	Song	Artist	Clean Score	Clean Grade
0	EASY ON ME	ADELE	3	E
1	MONEY	LISA	1	E
2	STAY	THE KID LAROI + JUSTIN BIEBER	12	E
3	RECKLESS	MADISON BEER	12	E
4	INDUSTRY BABY	LIL NAS X + JACK HARLOW	1	E
5	I LOVE YOU SO	THE WALTERS	14	E
6	HERE'S YOUR PERFECT	JAMIE MILLER	8	E
7	HEAT WAVES	GLASS ANIMALS	9	E
8	THE FEELS	TWICE	4	E
9	ANGEL BABY	TROYE SIVAN	9	E
10	HAPPIER	OLIVIA RODRIGO	21	D
11	LIFE GOES ON	OLIVER TREE	2	E
12	NEED TO KNOW	DOJA CAT	3	E
13	INFERNO	SUB URBAN + BELLA POARCH	9	E
14	LOVE BACK	WHY DON'T WE	39	D
15	ONE NIGHT	GRIFF	11	E
16	DOUBLE TAKE	DHRUV	93	A
17	IT'S YOU	SEZAI RI	10	E
18	LOVERBOY	A-WALL	18	E
19	KNOW ME TOO WELL	NEW HOPE CLUB + DANNA PAOLA	17	E
20	KISS ME MORE	DOJA CAT + SZA	4	E
21	DANDELIONS	RUTH B	7	E
22	LEAVE BEFORE YOU LOVE ME	MARSHMELLO + JONAS BROTHERS	6	E
23	FUTURE GHOST	WEIRD GENIUS FT. VIOLETTE WAUTIER	47	C
24	DRIVE YOU HOME	JACKSON WANG + INTERNET MONEY	40	C
25	BAD HABITS	ED SHEERAN	11	E
26	PERMISSION TO DANCE	BTS	9	E
27	HURRICANE	KANYE WEST + LIL BABY + THE WEEKND	5	E

To achieve the objectives of the project, I created one final Python file that would display the results of the model in an easy to read format. This Python file uses previous methods to scrape the song list of the most recent top 30 songs from the radio “hitz.fm” website at “<https://hitz.com.my/charts/hitz-30-chart>” and then scrapes the lyrics before filtering the unusable data, tokenizing, padding and predicting the lyrics using the model. The scores are stored and graded using the standard grading system between A and F. as seen in figure , an image is generated to display the results in nan easy to understand format.



From the model I was also able to generate a wordcloud that came from the songs that was labelled with value “0”. The wordcloud was generated with the removal of stopwords to see the frequency of words that matter, the larger the word the more frequent the letter is seen from

the songs. From the wordcloud, we can see that there are a few clearly profane words that are large indicating a frequent use such as “nigga”, “fuck” and “bitch” however the largest words either hold no sentimental value, non-profane or even good such as “love”. This is most likely due to the fact that most songs written are a mixture between love, lust and violence. Songs with lyrics like “I love my bitch” is a clear example of how the lyrics is described as bad but the word “love” is good.

4.2 Discussion

During the research phase of this project, a survey was conducted to find more information related to topic.

Here are some of the survey findings:

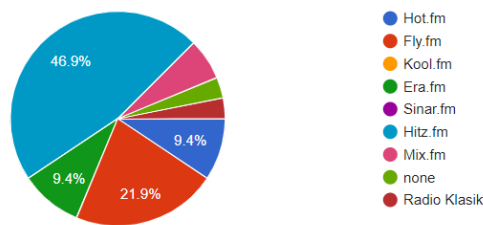


Figure 28: Pie Chart Representing Radio Channels People Listen to

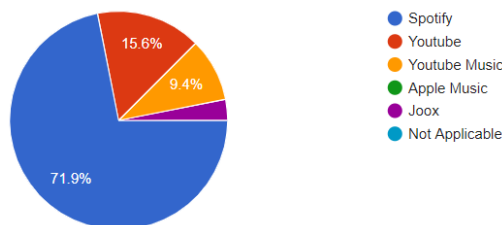


Figure 29 : Pie Chart Representing Streaming Services People Listen to

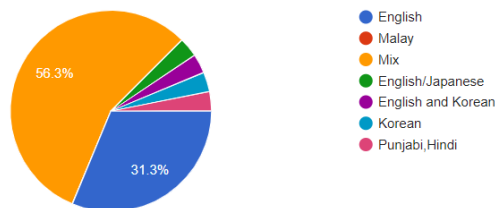


Figure 30: Pie Chart Representing Language of Songs People Listen to

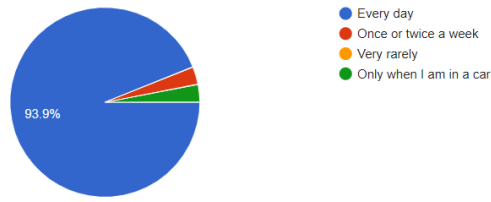


Figure 31: Pie Chart Representing the Frequency People Listen Music

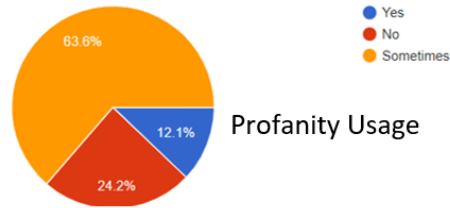


Figure 32: Pie Chart Representing How Much Profanity People Use

The above charts represent the most used streaming services, most visited radio channels as well as most listened languages that Malaysian youth listen to. The findings helped in minimizing the scope of the project to a more manageable scale

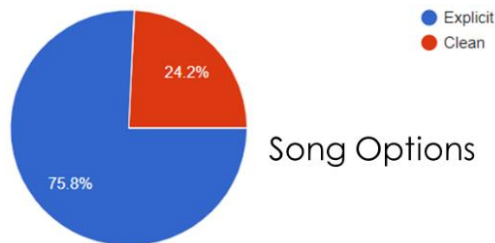


Figure 33: Pie Chart Representing Which Version of Songs People Prefer to Listen to

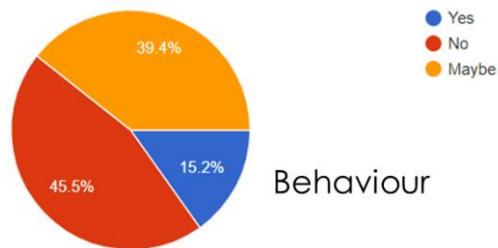


Figure 34: Pie Chart Representing People's Opinion of Music's Impact on Their Behaviour

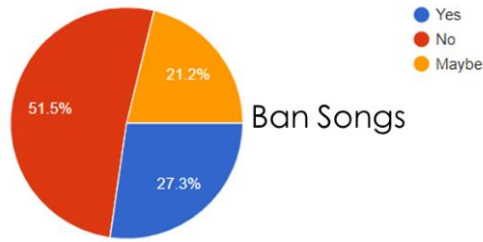


Figure 35: Pie Chart Representing People's Opinion on Banning Certain Songs

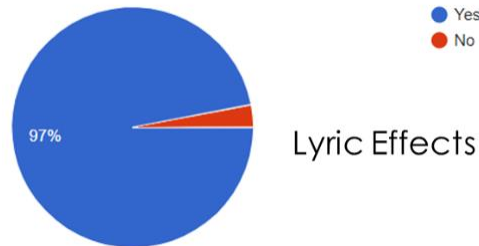


Figure 36: Pie Chart Representing People's Opinion of Lyrics Whether They Affect Their Fondness of The Song

The charts above describe the preferences and opinions of listeners. A large number of listeners prefer listening to explicit songs at the same time a large number of people cannot agree that music affects their behaviour indicating that most people are not aware or unconvinced of the effects of constant frequent exposure. Most people agree that their fondness towards a song is heavily affected by the lyrics and still many chose to listen explicit versions of songs over the clean ones. Many respondents do not agree with a “too strong” of a censorship of songs.

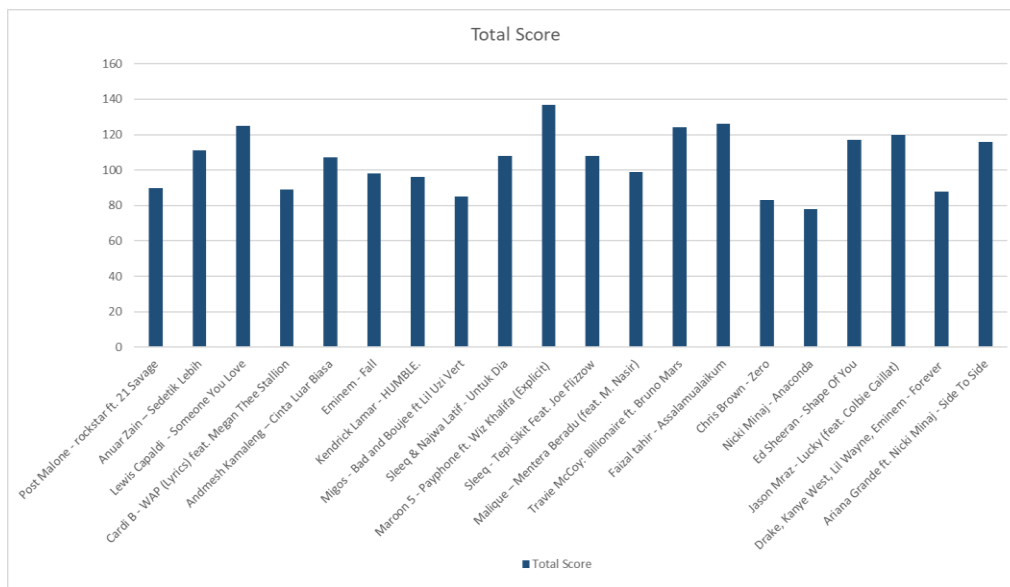


Figure 37: Bar Graph of Total Score 20 Songs Rated By Respondents

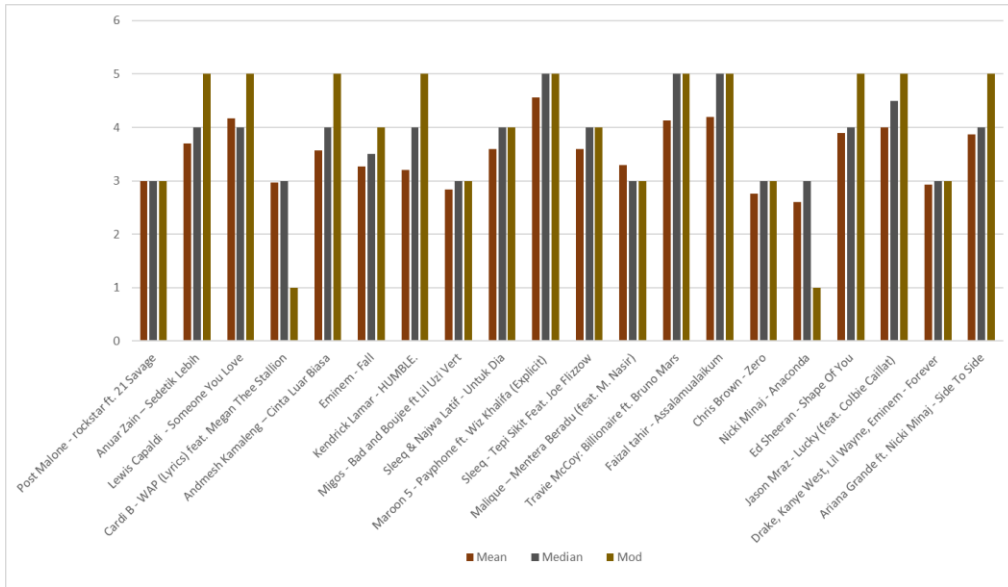


Figure 38: Bar Graph of Mean, Median and Mod Scores of Songs Rated By Respondents

The above two score charts indicate that although the survey respondents rated “dirty” songs lower than the clean ones, they do not fall low enough to be dismissed on a ranking list. Some of the explicit songs have even produced high results further supporting the idea that song lyrical content should be explored.

CHAPTER 5: CONCLUSION AND RECOMMENDATION

5.1 Conclusion

To conclude the project, this study has managed to achieve all three of its objectives as well as the success parameter set during the business understanding portion of this project. This project has managed to create its own dataset of popular songs and lyrics that Malaysians listen to and filter out the songs with a high volume of non-ASCII characters such as songs in Korean or Mandarin. The project has also succeeded in developing a model that is able to learn the patterns of both English and Malay languages which it then uses to categorize and score song lyrics based on “decency” or “cleanliness” scale from “0” to “1”. The project also managed to produce an easy to understand image that visualizes the songs and its decency level.

5.2 Recommendation

There are a few recommendations I would suggest to make this project better. The first is to have a bigger dataset of bad phrases in the Malay language. As of right now the Malay bad phrases I used was created using the bad phrases I know and found and it pales in comparison to the English bad words dataset. Perhaps with a bigger Malay profanity dataset, the prepared data could have a better labelling of which songs are bad and which songs are good in the Malay language.

Second, perhaps it is better to have a good phrases dataset to be included in the project. Perhaps instead of developing a model that learns by going through a whole songs and making decisions, it learns by going through phrases both bad and good. This way the model could more accurately determine the value of the song since it would be looking at the phrases instead of the whole song. The dataset could also be used in the data preparation phase, instead of simply scoring it “1” and “0” based on if it has profanity or not, the score could be assigned by calculating the percentage of bad phrases against the percentage of good phrases within a song and scoring it according to the larger percentage. If there are more good phrases, the label would be “1”, and if there are more bad phrases, the value would be “0”.

Third, prepare a better song dataset used for the training and testing of data. As of right now, the dataset I used for model training and testing was labelled using conditional statements. It would be better if someone were to create a dataset where people have actually graded the song as “clean” or “unclean” to be used instead of looking for key words. A method of doing this might include setting up a community platform where people can vote on songs they have

listened to and whether or not they think the song is appropriate for children's hearing. The resulting dataset would be a much better one than the one used in this project as the songs are labelled according to people who actually know the song.

Fourth, I recommend other websites that are more popular that have a song chart should be used to scrape Malaysia's most recent top rated songs as they might list out a more accurate finding of what Malaysians love to listen to. Perhaps it is possible to make a list from multiple websites and compile them together and give a new ranking based on the rankings from all the websites.

Lastly, it would be amazing if this project had was somehow integrated into social media websites such as "Twitter" or "Facebook" and hosted online. If the project were able to post its findings on the social media platforms, the public would be able to see and view the results and use it for themselves. Perhaps the project could publicise its finding on a weekly basis when the charts from different websites update themselves and people can decide whether or not they want their children to be exposed to certain songs.

References

- Chin, H., Kim, J., Kim, Y., Shin, J., & Yi, M. Y. (2018). Explicit Content Detection in Music Lyrics Using Machine Learning. *Proceedings - 2018 IEEE International Conference on Big Data and Smart Computing, BigComp 2018*, 517–521. <https://doi.org/10.1109/BigComp.2018.00085>
- DeFrank, M., & Kahlbaugh, P. (2019). Language Choice Matters: When Profanity Affects How People Are Judged. *Journal of Language and Social Psychology*, 38(1), 126–141. <https://doi.org/10.1177/0261927X18758143>
- Lan, T. S., & Logeswaran, R. (2020). Challenges and development in Malay natural language processing. *Journal of Critical Reviews*, 7(3), 61–65. <https://doi.org/10.31838/jcr.07.03.10>
- M. FRISBY, C., & BEHM-MORAWITZ, E. (2019). Undressing the Words: Prevalence of Profanity, Misogyny, Violence, and Gender Role References in Popular Music from 2006-2016. *Media Watch*, 10(1), 5–21. <https://doi.org/10.15655/mw/2019/v10i1/49562>
- Rahayu, B. A. (2018). Dangdut Music Affects Behavior Change at School and Adolescent Youth in Indonesia: A Literature Review. *Health Science Journal*, 12(1), 10–13. <https://doi.org/10.21767/1791-809x.1000552>
- Ying, O. J., Zabidi, M. M. A., Ramli, N., & Sheikh, U. U. (2020). Sentiment analysis of informal malay tweets with deep learning. *IAES International Journal of Artificial Intelligence*, 9(2), 212–220. <https://doi.org/10.11591/ijai.v9.i2.pp212-220>